UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/586,610	01/04/2007	Jacques Tisseau	293354US2X PCT	5964

22850 7590 05/17/2010
OBLON, SPIVAK, MCCLELLAND MAIER & NEUSTADT, L.L.P.
1940 DUKE STREET
ALEXANDRIA, VA 22314

EXAMINER	
PROCTOR, JASON SCOTT	

ART UNIT	PAPER NUMBER
2123	

NOTIFICATION DATE	DELIVERY MODE
05/17/2010	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patentdocket@oblon.com
oblonpat@oblon.com
jgardner@oblon.com

Office Action Summary	Application No.	Applicant(s)	
	10/586,610	TISSEAU ET AL.	
	Examiner	Art Unit	
	JASON PROCTOR	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 March 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *11-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *11-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Claims 11-20 were rejected in the Office Action entered on 28 August 2009.

Applicants' response submitted on 1 March 2010 has amended claims 11, 12, and 20. Claims 11-20 are pending in this application.

Claims 11-20 are rejected.

Specification

1. In response to the amendments to the specification, the previous objection to the specification is withdrawn.

Claim Rejections - 35 USC § 101

2. In response to the amendments to claim 11, the previous rejection of claims 11-20 under 35 U.S.C. § 101 are withdrawn.

Claim Rejections - 35 USC § 112

3. In response to the claim amendments and Applicants' remarks, the previous rejection of claims 11; 15 and 16; and 17, 18, and 19 are withdrawn.

Claim Objections

4. Claim 11 is objected to because of the following informalities: The present amendments introduce the phrase "…each of the chosen number of sequences…" in lines 23-24. To be

consistent with the claim terminology, it appears that this phrase should read "…each of the chosen number of times…" Appropriate correction or clarification is required.

5. Claim 12 is objected to because of the following informalities: The present amendments introduce the phrase "…each of including the designation of at least two state objects..." in lines 4-5. This phrase appears to contain a grammatical error. Appropriate correction is required.

Response to Remarks - 35 USC § 103

6. In response to the previous rejection of claims 11-20 under 35 U.S.C. § 103(a) as being unpatentable over Hiebeler in view of Johnson, Applicants argue primarily that:

> The objects described at page 5 of Hiebeler are asserted to teach state objects, and the agents described at page 5 of Hiebeler are asserted to teach interaction objects.
>
> First, the statement at page 5 of Hiebeler that agents represent entities in the model further clarifies that the Swarm system of Hiebeler is an entity-based simulation. Further, Hiebeler states that "[a]gents are the objects written by the user." However, Claim 11 defines "each interaction object including a designation of at least one of the state objects and of at least one function applicable to at least one of the state objects, and defining at each instant a topology of a system being simulated," rather than each interaction object being a state object written by a user. Thus, Hiebeler fails to teach state objects and interaction objects as defined by Claim 11.

The Examiner respectfully traverses this argument as follows.

The claim language includes no exclusion of "objects written by the user."

The claim recites "interaction objects" which are taught by Hiebeler's "agents":

> The claim recites that each interaction object includes "a designation of at least one of the state objects". Hiebeler teaches that agents contain a "data structure containing the internal state-variables for an agent" (Hiebeler, page 5).
>
> The claim recites that each interaction object additionally includes "a designation … of at least one function applicable to at least one of the state objects, and defining at each instant a topology of a system being simulated". Hiebeler teaches that agents contain a "step function, invoked on every time step (this is optional; users can write agents that have no internal dynamics, but only respond to message from other agents)" (Hiebeler, page 5).

Further related to both of the above limitations, Hiebeler teaches "Analysis objects" with "probe" and "set" messages. Hiebeler teaches that "A "probe" message is used to retrieve some internal state variable(s) from an agent... The "set" message is used to set an internal state variable in an object; parameters to the message are the name of the variable to set, and the new value to set it to." (Hiebeler, page 8). Regarding the "topology" feature, Hiebeler teaches that the spatial variables and/or positions of the agents define the topology of the system being simulated [*"The most commonly used analysis object is the* Image *object... Typically the image represents the values of spatial variable(s) as well as the positions of agents within a two-dimensional space."* (Hiebeler, page 8)].

Therefore Hiebeler appears to teach the claimed "interaction objects".

7. Applicants further argue that:

> Johnson, at the description of the scheduler and elsewhere, is silent regarding the above-quoted features of amended Claim 11. For example, even if, *arguendo*, an ActionGroup teaches a set of selected interaction objects, Johnson does not describe that the order of the ActionGroup is varied in a partly random manner for each of a chosen number of times.

The Examiner respectfully traverses this argument as follows.

Johnson expressly teaches that the order of the ActionGroup is varied in a partly random manner [*"The actions that go on in a simulation are orchestrated by objects that respond to the* Schedule *protocol."* (Johnson, page 64); *"The first three lines in the method create the* Schedule *named* modelSchedule... *Between the* createBegin *and* createEnd *methods, the only detail that this* Schedule *sets is the repeat interval, which is one. That means that all of the actions assigned to the* modelSchedule *will be executed each time step."* (Johnson, page 64); *An* ActionGroup *is a set of actions that are supposed to happen in sequence. The* buildActions *method is often designed to first create an* ActionGroup *and then to schedule that it is to be repeated every now and then… // One time tick, a set of several actions: **// randomize the order of agent updates (to be fair) … shuffler message: M(shuffleList :)…**"* (Johnson, pages 68-69)].

8. Applicants further argue that:

> Further, because the entity-based Swarm system of Hiebeler and Johnson does not teach state objects and interaction objects as defined by Claim 11 at all, Johnson cannot teach the simulation manager defined by amended Claim 11 to select interaction objects.

This argument has been traversed as shown above.

Regarding dependent claims 12-20, Applicants refer to the arguments addressed above.

Applicants' arguments have been fully considered but have been found unpersuasive.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. § 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. § 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was

made in order for the examiner to consider the applicability of 35 U.S.C. § 103(c) and potential 35 U.S.C. § 102(e), (f) or (g) prior art under 35 U.S.C. § 103(a).

9. Claims 11-20 are rejected under 35 U.S.C. § 103(a) as being unpatentable over "The Swarm Simulation System and Individual-based Modeling" by David Hiebeler ("Hiebeler") in view of "Swarm User Guide" by Paul Johnson et al. ("Johnson").

Regarding claim 11, Hiebeler teaches a device including a computer-readable storage medium storing computer-executable instructions therein, for simulating the real world, configured to be implanted in a computer and, when the computer-executable instructions are executed by a processor, cause the computer to perform interaction-based real world simulation [*"Swarm is a simulation environment which facilitates development and experimentation with simulations involving a large number of agents behaving and interacting within a dynamic environment."* (Hiebeler, abstract); *In the most general sense, Swarm is intended to provide an environment that facilitates the development of simulations involving a number of agents which exist within some (possibly dynamic) environment. This environment may be a regular spatial environment, a non-spatial environment such as a well-stirred soup, or something more abstract such as a telecommunications network. The agents communicate with each other and with the environment via messages"* (Hiebeler, page 4)], the device comprising:

state objects, each state object comprising at least one spatial or time data item or at least one property data item, defining a current state [*"An object in Swarm has three main*

characteristics: Name, Data, *and* Rules... *The Data are whatever local data the user wants to have in the agent (e.g. internal state variables)."* (Hiebeler, page 5)];

interaction objects, each interaction object including a designation of at least one of the state objects and of at least one function applicable to at least one of the state objects, and defining at each instant a topology of a system being simulated [*"The Rules are a set of functions that handle any messages that are sent to the object, including the "step" message."* (Hiebeler, page 5); *"Agents are the objects written by the user... When users write code for a new type of agent, there are several things that they must supply: ... A* step *function, invoked on every time step (this is optional; users can write agents that have no internal dynamics, but only respond to messages from other agents); Action functions that handle messages sent to the agent by other objects."* (Hiebeler, page 5)].

Johnson teaches a simulation manager configured to sequentially select, for a chosen number of times, each of a set of selected interaction objects to operate on each of a set of selected state objects based on a corresponding function, [*"The actions that go on in a simulation are orchestrated by objects that respond to the* Schedule *protocol."* (Johnson, page 64); *"The first three lines in the method create the* Schedule *named* modelSchedule... *Between the* createBegin *and* createEnd *methods, the only detail that this* Schedule *sets is the repeat interval, which is one. That means that all of the actions assigned to the* modelSchedule *will be executed each time step."* (Johnson, page 64)] wherein

an order by which the set of selected interaction objects is sequentially selected is varied in a partly random manner for each of the chosen number of times [*"An* ActionGroup *is a set of*

actions that are supposed to happen in sequence. The buildActions *method is often designed to first create an* ActionGroup *and then to schedule that it is to be repeated every now and then... // One time tick, a set of several actions: // randomize the order of agent updates (to be fair) ... shuffler message: M(shuffleList :)..."* (Johnson, pages 68-69)];

each of the set of selected interaction objects is selected only once for each of the chosen number of sequences [*selections*] [*"The actions that go on in a simulation are orchestrated by objects that respond to the* Schedule *protocol."* (Johnson, page 64); Example: *"In this simple example, the* modelSchedule *has only a single action, which instructs the one bug in the simulation, whose name is* aBug*, to carry out its method called step. It might be that there is a whole list of bugs,* bugList*, and each bug has to be instructed to carry out its step action. In such a case, the command would be...*" (Johnson, pages 64-65)]; and

each of the corresponding functions of each of the set of selected interaction objects is applied to a current state of each of the set of selected state objects and the current state of each of the set of selected state objects is evolved from a previous state based on a previous application of a corresponding function [*"The actions that go on in a simulation are orchestrated by objects that respond to the* Schedule *protocol."* (Johnson, page 64); *"The first three lines in the method create the* Schedule *named* modelSchedule... *Between the* createBegin *and* createEnd *methods, the only detail that this* Schedule *sets is the repeat interval, which is one. That means that all of the actions assigned to the* modelSchedule *will be executed each time step."* (Johnson, page 64); See also the example shown on page 69, which performs **at each time tick** the functions to **update all the agents** and kill off the agents who just died.].

Hiebeler and Johnson are analogous art because they are directed to the same type of simulation software.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Hiebeler and Johnson since Hiebeler expressly suggests that *"The current system uses a discrete time-stepped scheduling algorithm - on each time step, every object in the system receives a 'step' message, directing the agent to perform some unit of computation. This scheduling mechanism could be easily modified in some ways, for example to perform asynchronous random updating of objects; the next version of Swarm will provide even more flexible scheduling algorithms."* (Hiebeler, page 4). Therefore Hiebeler expressly suggests that an advantageous flexible scheduling system has been conceived and would be obvious to combine with the disclosed Swarm system. The Johnson reference describes a later version of the same Swarm system that implements that advantageous flexible scheduling system. The combination formed in this rejection is suggested and taught by the prior art as shown above.

Therefore it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Hiebeler and Johnson to arrive at the invention specified in claim 11.

Regarding claim 12, Hiebeler teaches that the simulation software comprises internal interaction objects, each including designation of a single state object and at least one function applicable to the single state object, and mutual interaction objects, each including the designation of at least two state objects and at least one function applicable to property data of

the designated at least two state objects [*"The Rules are a set of functions that handle any messages that are sent to the object, including the "step" message."* (Hiebeler, page 5); *"Agents are the objects written by the user… When users write code for a new type of agent, there are several things that they must supply: ... A* step *function, invoked on every time step (this is optional; users can write agents that have no internal dynamics, but only respond to messages from other agents); Action functions that handle messages sent to the agent by other objects."* (Hiebeler, page 5)].

Regarding claim 13, Hiebeler teaches that the simulation software is configured to modify at least some of the functions according to at least one property data item of at least one associated state object [*"Rather than building assumptions into Swarm about the type of environment agents would move around in, the notion of space was encapsulated within object. Currently, almost all of our sample experiments use a two-dimensional lattice space, implemented within an object we call GridSpace2, although other types of spaces are possible, such as: GridSpaceN, a general N-dimensional lattice. SoupSpace, in which agents meet/collide randomly. This would correspond to non-spatial models which assume thorough mixing. GraphSpace, an arbitrarily-connected graph describing which spatial sites are 'neighbors'... The space keeps track of the locations of any agents that are located in the space. When an agent wishes to move to a new location, it sends a request to space indicating where it wants to move; space replies, telling the agent where it actually ended up."* (Hiebeler, page 6)].

Regarding claim 14, Hiebeler teaches that the simulation software is configured to select at least some of the functions according to at least one property data item of at least one associated state object [*"Rather than building assumptions into Swarm about the type of environment agents would move around in, the notion of space was encapsulated within object. Currently, almost all of our sample experiments use a two-dimensional lattice space, implemented within an object we call GridSpace2, although other types of spaces are possible, such as: GridSpaceN, a general N-dimensional lattice. SoupSpace, in which agents meet/collide randomly. This would correspond to non-spatial models which assume thorough mixing. GraphSpace, an arbitrarily-connected graph describing which spatial sites are 'neighbors'... The space keeps track of the locations of any agents that are located in the space. When an agent wishes to move to a new location, it sends a request to space indicating where it wants to move; space replies, telling the agent where it actually ended up."* (Hiebeler, page 6)].

Alternatively, see Johnson, pages 68-69, teaching that an "ActionGroup" is a set of actions (i.e. functions to be executed), wherein the ActionGroup itself is a property data item of some associated state object.

Regarding claim 15, Hiebeler teaches that at least some of the state objects comprise a property data item representing an intensive variable [*"An object in Swarm has three main characteristics:* Name, Data, *and* Rules... *The Data are whatever local data the user wants to have in the agent (e.g. internal state variables)."* (Hiebeler, page 5)].

Regarding claim 16, Hiebeler teaches that at least some of the interaction objects have a function bringing about an extensive or intensive variable [*"An object in Swarm has three main characteristics:* Name, Data, *and* Rules... *The Data are whatever local data the user wants to have in the agent (e.g. internal state variables)."* (Hiebeler, page 5)].

Regarding claim 17, Johnson teaches that at least some of the state objects comprise state sub-objects [*"Object oriented programming (OOP) is well suited to describe autonomous agents, so it should have appeal to scientists and modelers on that basis alone... The features we emphasize here are encapsulation and inheritance."* (Johnson, page 18)].

Regarding claim 18, Johnson teaches that at least some of the state objects comprise interaction sub-objects operating on the said state sub-objects [*"The objects that represent the actors in a simulation - the substantively important entities - are usually subclassed from the* SwarmObject *class. The 'inheritance hierarchy' that leads to the class* SwarmObject *passes through classes that allow the creation and deletion of objects from a simulation."* (Johnson, page 29); The claim appears to be referring to class inheritance, a well-known concept taught by Johnson at the portion cited and elsewhere.].

Regarding claim 19, Johnson teaches that the simulation software comprises classes of objects defining structures of state objects and of interaction objects, the state objects and interaction objects being derived from these classes by instancing [*"Objects are created through*

a process called 'instantiation.' Put tersely, code is written in 'classes' and then objects are created as instances of their classes." (Johnson, page 17)].

Regarding claim 20, Johnson teaches that the simulation software comprises a scheduler operating according to one of two modes selected from a real-time mode, in which it operates according to a selected frequency, and a virtual-time mode in which it operates periodically but for durations which vary from one period to another [*"A Swarm simulation proceeds in discrete time steps."* (Johnson, page 19)].

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached at (571) 272-3753. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Jason Proctor/
Primary Examiner, Art Unit 2123

jsp